

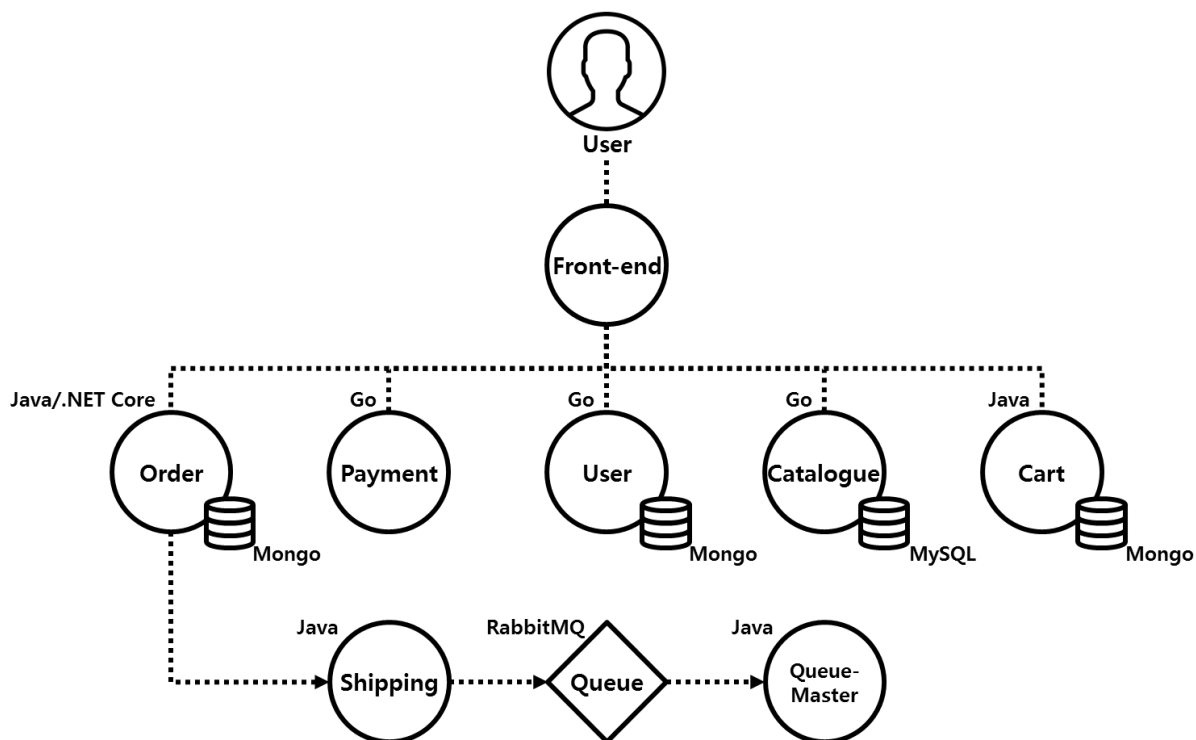
ZCP Private 성능 Test

1 Sample App.

1.1 Sock shop

MSA 를 활용한 Online Shopping Mall Demo 로 Java, Go, NodeJS, MongoDB(NoSQL), MySQL(RDBMS), RabbitMQ 등을 사용하며 가장 흔하게 사용되는 MSA Application.

1.2 Sample App. Architecture

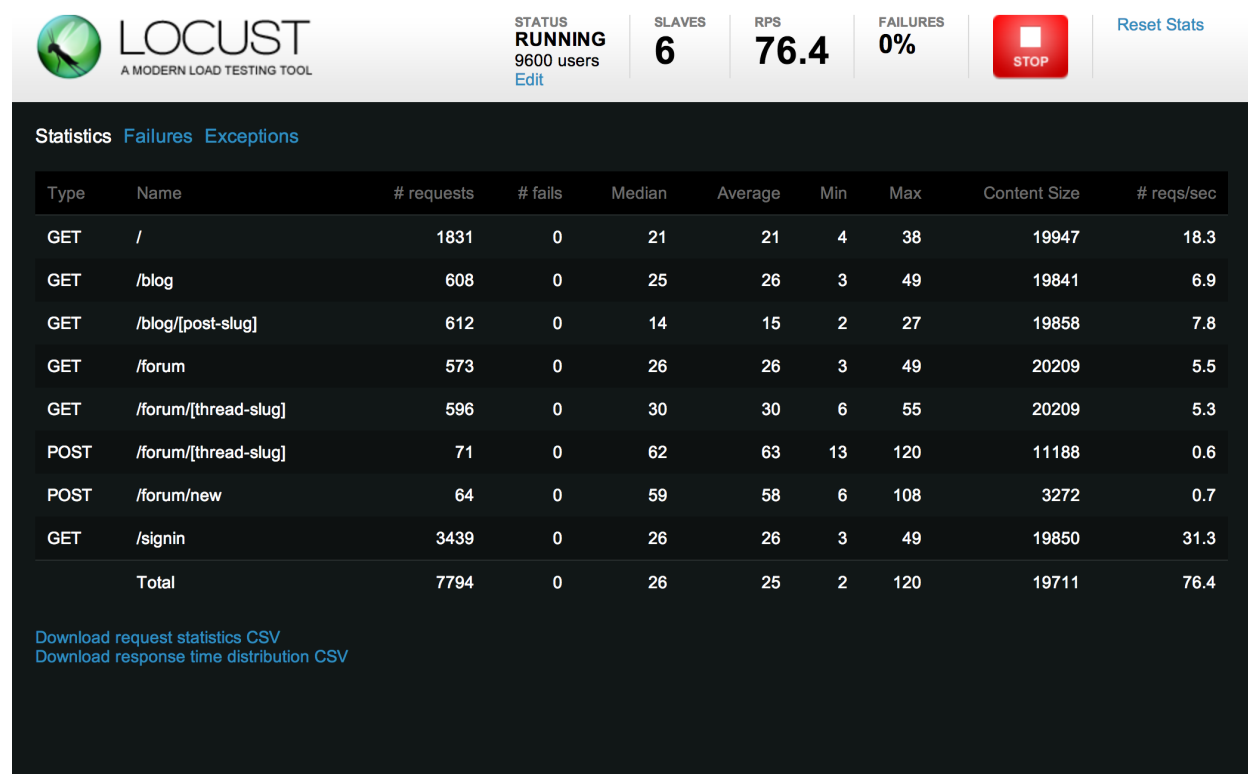


각각의 서비스들은 Docker Container 로 구성

2 Load Testing Tool

2.1 Locust

Python 기반의 Open Source Load Testing Tool 로 Python 스크립트를 통해 테스트 시나리오 구성되며 거의 모든 시스템에 대해 부하 테스트 가능.



3 Installation

3.1 Sample App. Installation

```
$ git clone https://github.com/microservices-demo/microservices-demo.git
$ kubectl create namespace sock-shop
$ cd microservices-demo/deploy/kubernetes/
$ kubectl create -f complete-demo.yaml
```

3.2 Load Testing Tool Installation

```
$ cd ..  
$ git clone https://github.com/RGKim/Locust_k8s.git  
$ cd Locust_k8s  
$ kubectl create namespace locust  
$ kubectl create configmap locustd-configmap --from-file=locustfile.py -n locust  
$ sed -i "s/TARGETIP/<SAMPLE APP's IP>:80/g" /k8s/locust-master.yaml  
$ sed -i "s/TARGETIP/<SAMPLE APP's IP>:80/g" /k8s/locust-slave.yaml  
$ kubectl create -f /k8s/. -n locust
```

4 측정 방법

4.1 Test Scenario

다음의 Scenario 를 locustfile.py 로 작성하여 구성

1. **catalogue Pod 에서 랜덤으로 하나의 item 선정**
 - A. GET /catalogue
 - B. catalogue POD 호출
 - C. catalogue-db POD 호출
2. **접속 페이지 호출**
 - A. GET /
 - B. front-end POD 호출
3. **login**
 - A. GET /login
 - B. user POD 호출
 - C. user-db POD 호출
4. **category 페이지 호출**
 - A. GET /category.html
 - B. front-end POD 호출
 - C. 1.에서 선정 된 item 에 대한 detail 페이지 호출
 - D. GET /detail.html?id={}
 - E. front-end POD 호출

5. cart 삭제(혹시 아이템이 들어 있을 경우 방지)

- A. DELETE /cart
- B. carts POD 호출
- C. carts-db POD 호출

6. cart 에 item 담기

- A. POST /cart
- B. carts POD 호출
- C. carts-db POD 호출

7. basket 페이지 호출

- A. GET /basket.html
- B. front-end POD 호출

8. 주문

- A. POST /orders
- B. orders POD 호출
- C. payment POD 호출
- D. shipping POD 호출
- E. rabbitmq POD 호출
- F. queue-master POD 호출
- G. orders-db POD 호출